



TUGAS AKHIR – TE 141599

**PERANCANGAN *FUZZY MODIFIED MODEL*
REFERENCE ADAPTIVE CONTROL UNTUK
PENGATURAN KECEPATAN MOTOR ARUS
SEARAH TANPA SIKAT**

Muhammad Iqbal Fauzi
NRP. 2211 100 021

Dosen Pembimbing
Ir. Rusdhianto Effendie A.K., M.T.
Eka Iskandar S.T., M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT – TE 141599

***DESIGN OF FUZZY MODIFIED MODEL REFERENCE
ADAPTIVE CONTROL FOR SPEED CONTROL OF
BRUSHLESS DIRECT CURRENT MOTOR***

Muhammad Iqbal Fauzi
NRP. 2211 100 021

Supervisor
Ir. Rusdhianto Effendie A.K., M.T.
Eka Iskandar S.T., M.T.

***DEPARTEMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Sepuluh Nopember Insitute of Technology
Surabaya 2016***

**PERANCANGAN FUZZY MODIFIED MODEL REFERENCE
ADAPTIVE CONTROL UNTUK PENGATURAN KECEPATAN
MOTOR ARUS SEARAH TANPA SIKAT**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I



Ir. Rusdhianto Effendie A.K., M.T.

NIP. 195704241985021001

Dosen Pembimbing II



Eka Iskandar, S.T., M.T.

NIP. 198005282008121001



PERANCANGAN *FUZZY MODIFIED MODEL REFERENCE ADAPTIVE CONTROL* UNTUK PENGATURAN KECEPATAN MOTOR ARUS SEARAH TANPA SIKAT

Muhammad Iqbal Fauzi
2211 100 021

Dosen Pembimbing I : Ir. Rusdhianto Effendie A.K., MT.
Dosen Pembimbing II : Eka Iskandar, ST., MT.

ABSTRAK

Di dalam tugas akhir ini, saya merancang sebuah sistem kendali kecepatan motor BLDC menggunakan *Model Reference Adaptive Control* (MRAC) dimodifikasi dengan *fuzzy inference system*. MRAC yang dirancang menggunakan mekanisme penyesuaian PI. Performa sistem diperbaiki lebih jauh lagi menggunakan *fuzzy inference system* yang akan memodifikasi keluaran kontroler berdasarkan *error* antara *output plant* dan *output* model. Setelah dilakukan simulasi, didapatkan bahwa selisih waktu *rise time* Fuzzy-MRAC terhadap model referensi sebesar 0,0147 untuk beban maksimal dan 0,0228 untuk beban minimal dengan RMSE sebesar 2,6319 dan 3,6207 untuk masing-masing beban.

Kata Kunci : *Brushless DC, Fuzzy, Model Reference Adaptive*

**DESIGN OF FUZZY MODIFIED MODEL REFERENCE
ADAPTIVE CONTROL FOR SPEED CONTROL OF BRUSHLESS
DIRECT CURRENT MOTOR**

Muhammad Iqbal Fauzi
2211 100 021

Supervisor I : Ir. Rusdhianto Effendie A.K., MT.
Supervisor II : Eka Iskandar, ST., MT.

ABSTRACT

In this final project, I designed a BLDC motor speed control system using Model Reference Adaptive Control (MRAC) modified by fuzzy inference system. Designed MRAC used PI adjustment mechanism. The system's performance was improved further by the use of fuzzy inference system that modified the controller output based on the error between plant and model. After the simulation, it brings that rise time difference between Fuzzy-MRAC and model reference is 0.0147 for maximum load and 0.0228 for minimum load with RMSE of 2.6319 and 3.6207 respectively.

Keywords : Brushless DC, Fuzzy, Model Reference Adaptive

KATA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kehadiran Allah SWT karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan penulisan buku tugas akhir dengan judul **“PERANCANGAN FUZZY MODIFIED MODEL REFERENCE ADAPTIVE CONTROL UNTUK PENGATURAN KECEPATAN MOTOR ARUS SEARAH TANPA SIKAT”**. Tugas akhir merupakan salah satu syarat yang harus dipenuhi untuk menyelesaikan program studi Strata-1 pada Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menyadari bahwa dalam penulisan tugas akhir ini banyak mengalami kendala, namun berkat bantuan, bimbingan, dan kerja sama dari berbagai pihak sehingga kendala-kendala tersebut dapat diatasi. Untuk itu pada kesempatan ini penulis menyampaikan banyak terimakasih dan penghargaan setinggi-tingginya kepada :

1. Kedua orang tua, Sunardi dan Ibunda Faridah yang selalu memberikan dukungan, semangat, dan doa kepada penulis.
2. Bapak Rusdhi dan Bapak Eka selaku Dosen Pembimbing atas segala bantuan, perhatian, dan arahan selama pengerjaan tugas akhir ini.
3. Rekan-rekan e51 khususnya bidang studi Sistem Pengaturan.
4. Teman-teman asisten lab B405.

Penulis berharap tugas akhir ini dapat bermanfaat bagi yang membutuhkannya.

Surabaya, 11 Januari 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
LEMBAR PENGESAHAN	vii
ABSTRAK.....	i
ABSTRACT.....	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Sistematika Penulisan	2
1.6 Relevansi.....	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 Motor <i>Brushless</i> DC	5
2.1.1 Konstruksi Motor <i>Brushless</i> DC	5
2.2 Rem Elektromagnetik	9
2.3 Rangkaian <i>Optocoupler</i>	10
2.3.1 Rangkaian <i>Voltage Follower</i>	10
2.4 Identifikasi Sistem	11
2.5 <i>Fuzzy Logic</i>	12
2.5.1 <i>Fuzzy Set dan Membership Function</i>	12
2.5.2 <i>Fuzzy If-Then Rule</i>	13
2.5.3 <i>Sugeno Fuzzy Model</i>	13
2.6 <i>Model-Reference Adaptive System</i>	13
2.6.1 MRAS berbasis <i>Output Feedback</i>	14
2.7 <i>Fuzzy Modified</i> MRAC	18
BAB 3 PERANCANGAN SISTEM	19
3.1 Gambaran Umum Sistem	19
3.2 Perancangan Perangkat Keras	19
3.2.1 Perancangan Mekanik	19
3.2.2 Perancangan Elektrik	20

3.3	Perancangan Perangkat Lunak	21
3.3.1	Perangkat Lunak Arduino	21
3.4	Perangkat Lunak Matlab	22
3.5	Identifikasi dan Pemodelan Sistem	23
3.6	Perancangan <i>Fuzzy</i> -MRAC	24
3.6.1	Perancangan MRAC	24
3.6.2	Perancangan Kontroler <i>Fuzzy</i>	25
BAB 4 PENGUJIAN DAN ANALISA		29
4.1	Gambaran Umum Pengujian Sistem	29
4.2	Pengujian Sensor Kecepatan	29
4.3	Pengujian <i>Open Loop</i> Kecepatan Motor	30
4.4	Simulasi Sistem	31
4.4.1	Blok Diagram Simulink	31
4.4.2	Hasil dan Analisa Simulasi	35
4.5	Implementasi Sistem	40
4.5.1	Realisasi <i>Plant</i>	40
4.5.2	Implementasi Kontroler pada <i>Plant</i> P-1	41
BAB 5 PENUTUP		45
4.6	Kesimpulan	45
4.7	Saran	45
DAFTAR PUSTAKA		47
LAMPIRAN		49

TABLE OF CONTENT

ABSTRAK.....	ix
ABSTRACT.....	xi
PREFACE	xiii
TABLE OF CONTENT	xv
LIST OF ILLUSTRATIONS	xvii
LIST OF TABLE	xix
BAB 1 INTRODUCTION	1
1.1 Background	1
1.2 Problem Formulation	1
1.3 Scope of Problem	2
1.4 Research Purpose	2
1.5 Writing Sistematic	2
1.6 Relevance	3
BAB 2 LITERATURE.....	5
2.1 Brushless DC Motor	5
2.1.1 Brushless DC Motor Construction.....	5
2.2 Electromagnetic Brake	9
2.3 Optocoupler Circuit	10
2.3.1 Voltage Follower Circuit	10
2.4 System Identification	11
2.5 Fuzzy Logic	12
2.5.1 Fuzzy Set and Membership Function.....	12
2.5.2 Fuzzy If-Then Rule	13
2.5.3 Sugeno Fuzzy Model	13
2.6 Model Reference Adaptive System	13
2.6.1 Output Feedback-based MRAS.....	14
2.7 Fuzzy Modified MRAC	18
BAB 3 SYSTEM DESIGN	19
3.1 System Overview	19
3.2 Hardware Design	19
3.2.1 Mechanic Design	19
3.2.2 Electric Design.....	20
3.3 Software Design.....	21
3.3.1 Arduino Software.....	21
3.4 Matlab Software.....	22

3.5	System Identification and Modelling	23
3.6	Fuzzy-MRAC Design.....	24
3.6.1	MRAC Design.....	24
3.6.2	Fuzzy Controller Design	25
BAB 4	EXAMINATION AND ANALYSIS	29
4.1	System Testing Overview	29
4.2	Speed Sensor Testing	29
4.3	Open Loop Motor Speed Testing	30
4.4	System Simulation.....	31
4.4.1	Simulink Block.....	31
4.4.2	Simulation Result and Analysis	35
4.5	System Implementation.....	40
4.5.1	Plant Realization	40
4.5.2	Controller Implementation on Plant P-1.....	41
BAB 5	CONCLUSION	45
5.1	Conclusion.....	45
5.2	Suggestion	45
BIBLIOGRAPHY	47
ENCLOSURE	49
BIOGRAPHY	51

DAFTAR GAMBAR

Gambar 2.1 Motor BLDC 3-Fasa dengan 4 <i>Pole</i> Rotor dan 12 Slot Stator.	6
Gambar 2.2 Gelombang <i>Back-emf</i> dan Arus Fasa untuk Motor BLDC Tiga Fasa dengan Arus Bipolar 120°.....	8
Gambar 2.3 Skematik dari <i>Inverter</i> Berbasis IGBT untuk Motor BLDC 3 Fasa	8
Gambar 2.4 Prinsip Arus <i>Eddy</i> Pada Logam yang Bergerak	9
Gambar 2.5 Rangkaian <i>Optocoupler</i>	10
Gambar 2.6 Rangkaian <i>Voltage Follower</i>	11
Gambar 2.7 Contoh Fungsi Keanggotaan	12
Gambar 2.8 Diagram Blok <i>Model Reference Adaptive System</i>	14
Gambar 2.9 Diagram Blok <i>Model-Reference Adaptive System</i>	17
Gambar 2.10 Blok Diagram <i>Fuzzy Modified MRAC</i>	18
Gambar 3.1 Pengkabelan Sistem Elektrik <i>Plant</i>	21
Gambar 3.2 Diagram Alir Program Arduino	22
Gambar 3.3 Diagram Blok Simulink untuk Identifikasi	23
Gambar 3.4 Fungsi Keanggotaan <i>Input</i> Kontroler <i>Fuzzy</i>	26
Gambar 4.1 Hubungan <i>Input-Output Plant</i>	30
Gambar 4.2 Blok Simulink	32
Gambar 4.3 Blok Simulink Filter.....	33
Gambar 4.4 Blok Simulink <i>Fuzzy</i>	34
Gambar 4.5 Blok Simulink <i>Adjustment</i>	35
Gambar 4.6 <i>Output Plant</i> Saat Adaptasi Sampai Detik ke 200.....	36
Gambar 4.7 Perubahan Parameter Kontroler	36
Gambar 4.8 <i>Step</i> Respon Saat Beban Nominal	37
Gambar 4.9 <i>Step</i> Respon Saat Beban Maksimal	38
Gambar 4.10 <i>Step</i> Respon Saat Beban Minimal	38
Gambar 4.11 <i>Error</i> Saat Beban Maksimal	39
Gambar 4.12 <i>Error</i> Saat Beban Minimal	39
Gambar 4.14 <i>Plant</i> motor BLDC P-1.....	41
Gambar 4.15 <i>Step</i> Respon Sistem Saat Implementasi pada Beban Minimal	42
Gambar 4.16 <i>Step</i> Respon Sistem Saat Implementasi pada Beban Nominal	43
Gambar 4.17 <i>Step</i> Respon Sistem Saat Implementasi pada Beban Maksimal	43

DAFTAR TABEL

Tabel 3.1 Daftar Warna Kabel <i>Input-Output Driver</i> Motor dan Fungsinya	20
Tabel 3.2 Fungsi <i>Transfer Plant</i> Hasil Identifikasi	24
Tabel 4.1 Hasil Pengujian Sensor Kecepatan.....	30
Tabel 4.2 Perbandingan Indeks Performansi Respon <i>Step</i> dari MRAC dan FMRAC pada Beban Maksimal	40
Tabel 4.3 Perbandingan Indeks Performansi Respon Transien dari MRAC dan FMRAC pada Beban Minimal.....	40
Tabel 4.4 Indeks Performansi Respon Transien MRAC dan FMRAC Pada Beban Minimal.....	44
Tabel 4.5 Indeks Performansi Respon Transien MRAC dan FMRAC Pada Beban Nominal	44
Tabel 4.6 Indeks Performansi Respon Transien MRAC dan FMRAC Pada Beban Maksimal	44

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Saat ini, motor arus searah tanpa sikat telah banyak menggantikan motor arus searah konvensional. Hal ini dikarenakan motor arus searah tanpa sikat memiliki performa yang lebih baik dibanding motor arus searah konvensional dilihat dari segi efisiensi, *lifetime*, *torque delivery*, dan operasi kecepatan tinggi [1]. Motor arus searah tanpa sikat telah diaplikasikan di berbagai aplikasi antara lain otomotif, *Heating Ventilating and Air Conditioning* (HVAC), mesin CNC, robot industri dan lain-lain.

Pada aplikasi *servo*, sistem pengaturannya memiliki parameter yang berubah-ubah selama beroperasi sehingga mempengaruhi respon sistem. Skema *Model Reference Adaptive Control* (MRAC) dikembangkan untuk meminimalkan pengaruh variasi parameter tersebut. Pada skema MRAC, perbedaan antara nilai *output plant* dan *output* model referensi diproses melalui suatu algoritma untuk mengubah nilai parameter kontroler sedemikian rupa sehingga respon *output plant* sama dengan respon *output* model referensi. Namun, untuk menyamakan respon tersebut memerlukan waktu yang cukup lama.

Fuzzy Logic menawarkan metode yang sederhana untuk merancang kontroler *non-linier* berdasarkan pendekatan heuristik [2]. Pada penelitian kali ini, dirancang sebuah sistem MRAC yang dimodifikasi *fuzzy inference system* model Sugeno. *Fuzzy inference system* digunakan untuk memperbaiki performa transien dari MRAC. Kontroler yang sudah dibuat kemudian diaplikasikan pada sebuah model motor BLDC dan disimulasikan menggunakan Matlab Simulink.

1.2 Perumusan Masalah

Permasalahan pada tugas akhir ini adalah lamanya waktu konvergensi dari MRAC. Untuk itu, dilakukan perancangan sebuah sistem *model reference adaptive control* yang dimodifikasi dengan *fuzzy inference system* sehingga waktu konvergensi antara *output plant* dengan *output* model referensi dapat dipercepat dibandingkan dengan MRAC biasa.

1.3 Batasan Masalah

Permasalahan pada tugas akhir ini dibatasi oleh beberapa hal antara lain:

- a. Permasalahan kontrol adalah permasalahan *tracking*.
- b. Tidak dibahas masalah komutasi motor BLDC.

1.4 Tujuan Penelitian

Tujuan dari pelaksanaan tugas akhir, antara lain:

- a. Mengidentifikasi kinematika dan dinamika motor arus searah tanpa sikat sehingga didapatkan model matematika sistem.
- b. Merancang *model reference adaptive control* untuk pengaturan kecepatan motor arus searah tanpa sikat.
- c. Merancang *model reference adaptive control* dimodifikasi *fuzzy inference system* untuk pengaturan kecepatan motor arus searah tanpa sikat.
- d. Menganalisa hasil simulasi untuk didapatkan perbandingan performa antara 2 strategi pengaturan di atas.

Hasil yang diperoleh dari pelaksanaan tugas akhir diharapkan dapat memberikan manfaat, antara lain :

- a. Dapat dijadikan acuan dalam pembuatan sistem pengaturan kecepatan motor arus searah tanpa sikat.
- b. Dapat dijadikan referensi penelitian lebih lanjut mengenai pengaturan kecepatan motor arus searah tanpa sikat.

1.5 Sistematika Penulisan

Buku tugas akhir ini terdiri dari lima bab dan disusun menurut sistematika penulisan berikut ini

BAB I: PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, sistematika penulisan, dan relevansi.

BAB 2: DASAR TEORI

Bab ini berisi tentang teori yang menunjang penelitian, berupa teori tentang BLDC dan komponennya, serta metode yang digunakan untuk pengaturan kecepatan motor BLDC.

BAB 3: PERANCANGAN SISTEM DAN KONTROLER

Bab ini berisi tentang perancangan perangkat keras, perangkat lunak, dan perancangan kontroler.

BAB 4: PENGUJIAN DAN ANALISA

Bab ini berisi tentang hasil simulasi kontroler dan analisisnya. Selain itu berisi tentang hasil implementasi kontroler pada Simulator BLDC beserta analisa hasil implementasi.

BAB 5: KESIMPULAN DAN SARAN

Berisi kesimpulan dan saran yang dapat dijadikan pertimbangan pengembangan berdasar hasil pengerjaan tugas akhir ini.

1.6 Relevansi

Hasil dari tugas akhir ini diharapkan dapat memberikan manfaat dalam pengembangan penelitian tentang BLDC khususnya strategi kontrol motor listrik dalam BLDC.

BAB 2

TINJAUAN PUSTAKA

2.1 Motor *Brushless* DC [3]

Motor *Brushless* DC (BLDC) adalah motor sinkron dengan magnet permanen pada rotor dan kumparan jangkar pada stator. Keuntungan paling jelas dari konfigurasi tanpa sikat adalah penghilangan sikat, yang menghilangkan pemeliharaan sikat dan bunga api. Penggunaan kumparan jangkar pada stator membantu konduksi panas pada kumparan. Karena tidak ada kumparan pada rotor, kerugian listrik dalam rotor menjadi minimal. Motor BLDC lebih disukai dibandingkan dengan motor induksi untuk daerah daya sebagian-sebagian.

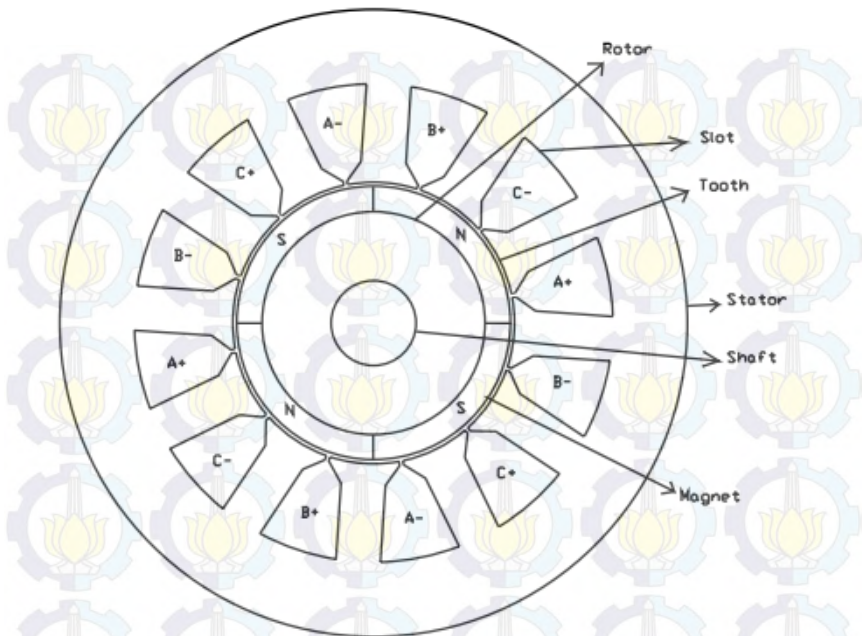
Motor BLDC memiliki efisiensi yang dan faktor daya yang lebih baik dan oleh karena itu, daya keluaran lebih besar untuk kerangka yang sama, karena penguatan medan dipengaruhi oleh magnet permanen dan tidak harus disuplai oleh arus jangkar. Keuntungan-keuntungan yang didapat dari motor BLDC disertai dengan peningkatan kompleksitas di kontroler elektronik dan perlunya penginderaan posisi poros. Eksitasi magnet permanen (PM) lebih layak untuk motor kecil, biasanya di bawah 20 kW. Untuk motor yang lebih besar, biaya dan berat magnet menjadi terlalu tinggi, dan itu akan membuat lebih masuk akal untuk memilih eksitasi dengan elektromagnet.

Namun, dengan pengembangan bahan magnet permanen medan tinggi, motor PM dengan kapasitas beberapa MW telah dibangun.

2.1.1 Konstruksi Motor *Brushless* DC

2.1.1.1 Magnet Permanen

Pada motor BLDC, magnet permanen dipasang pada rotor. Magnet dibangun dalam bentuk melengkung dan ditempel pada permukaan rotor dengan kutub yang berlawanan memiliki polaritas magnet yang berbeda seperti ditunjukkan pada Gambar 2.1.



Gambar 2.1 Motor BLDC 3-Fasa dengan 4 *Pole* Rotor dan 12 *Slot* Stator.

2.1.1.2 Gulungan Stator

Motor BLDC sering dianggap memiliki tiga fasa, tetapi hal ini tidak selalu terjadi. Motor kecil untuk aplikasi ringan seperti kipas pendingin lebih murah pembiayaannya jika dibangun dengan satu atau dua fasa. Di sisi lain, pada aplikasi berat dengan *rating* MW, akan lebih baik jika menggunakan motor dengan jumlah fasa yang banyak. Motor dengan 15 fasa telah dibangun untuk propulsi kapal.

Jumlah stator dipilih tergantung pada kutub-kutub rotor, jumlah fasa, dan konfigurasi gulungan. Secara umum, desain *slot*/kutub fraksional banyak dipilih untuk meminimalkan torsi *cogging*.

2.1.1.3 Karakteristik Motor

Bentuk gelombang *air-gap-flux-density* pada dasarnya adalah sebuah gelombang persegi, tapi *fringing* menyebabkan bentuk gelombangnya menjadi sedikit melengkung. Saat rotor berputar, bentuk

gelombang tegangan yang terinduksi pada tiap fasa terhadap waktunya merupakan replika dari bentuk gelombang *air-gap-flux-density* terhadap posisi rotor. *Fringing* menyebabkan bentuk gelombang *back-emf* berbentuk *trapezoidal*. Bentuk gelombang *trapezoidal* inilah yang membedakan motor BLDC dengan *permanent magnet synchronous motor* (PMSM), yang memiliki bentuk gelombang *back-emf* berupa sinusoidal. Dengan memberikan arus *rectangular* pada tiap fasa yang *back-emf*-nya berada pada keadaan nilai penuh, dimungkinkan untuk mendapatkan torsi motor BLDC yang hampir konstan.

Tegangan *back-emf* dan gelombang arus fasa 120° ideal untuk motor BLDC tiga fasa ditunjukkan pada Gambar 2.2. *Switch inverter* yang aktif di tiap interval 60° ditunjukkan pada Gambar 2.3.

Amplitudo dari tegangan *back-emf* sebanding dengan kecepatan rotor dengan Persamaan 2.1

$$E = k\phi\omega_m \quad (2.1)$$

Di mana k adalah konstanta yang tergantung pada jumlah lilitan di tiap fasa, ϕ adalah fluks magnet permanen, dan ω_m adalah kecepatan mekanis.

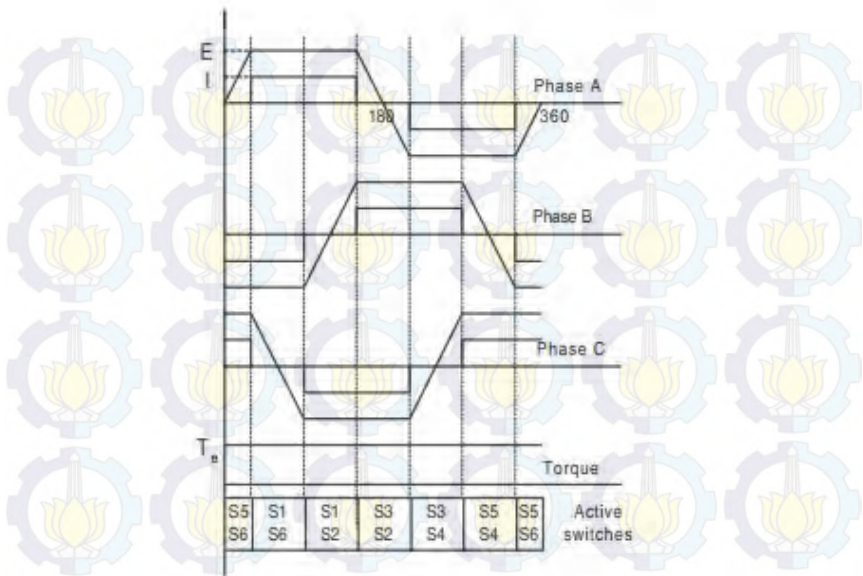
Selama tiap interval 120° , daya sesaat yang dikonversi dari elektris ke mekanis adalah

$$P_o = \omega_m T_e = 2EI \quad (2.2)$$

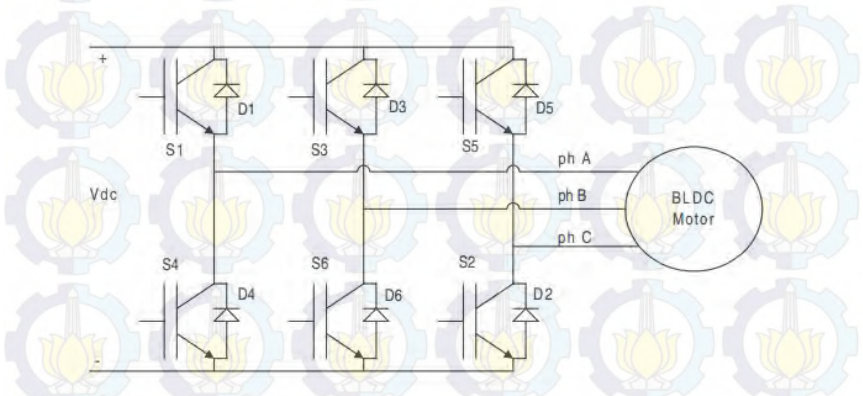
Di mana T_e adalah torsi keluaran dan I adalah amplitudo arus fasa. Dari Persamaan 2.1 dan 2.2, persamaan torsi keluaran dapat ditulis

$$T_e = 2k\phi I = k_t I$$

Di mana k_t adalah konstanta torsi.



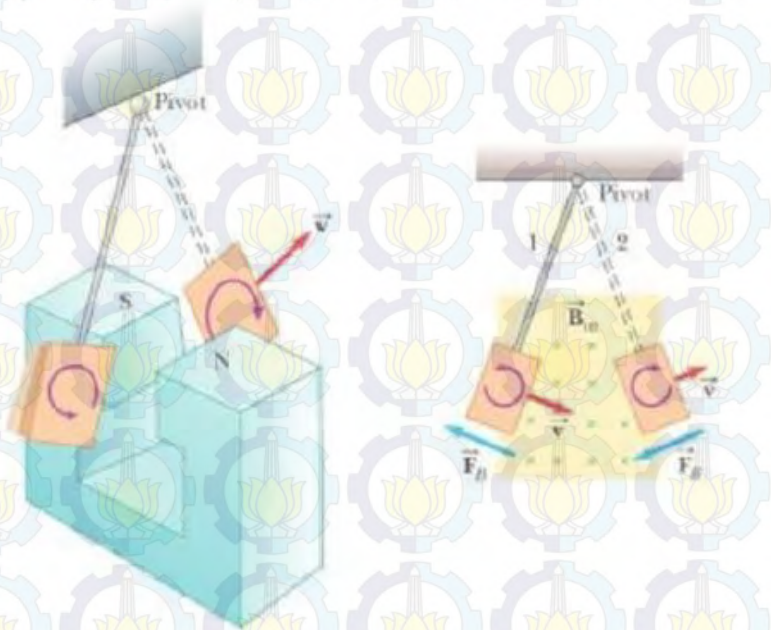
Gambar 2.2 Gelombang *Back-emf* dan Arus Fasa untuk Motor BLDC Tiga Fasa dengan Arus Bipolar 120°



Gambar 2.3 Skematik dari *Inverter* Berbasis IGBT untuk Motor BLDC 3 Fasa

2.2 Rem Elektromagnetik [4]

Sistem pengereman ini menggunakan gaya elektromagnetik yang timbul dari suatu magnet permanen tetap atau kumparan yang diberikan arus listrik untuk memperlambat suatu gerakan. Konstruksi dasarnya berupa suatu piringan logam *non-feromagnetik* yang terpasang pada suatu poros yang berputar. Piringan logam tersebut diapit oleh kumparan yang dialiri arus listrik hingga menimbulkan medan magnet yang kutubnya saling berlawanan. Logam piringan tersebut akan memotong medan magnet yang ditimbulkan oleh kumparan tersebut sehingga menimbulkan *eddy current* atau arus *eddy*.



Gambar 2.4 Prinsip Arus *Eddy* Pada Logam yang Bergerak [5]

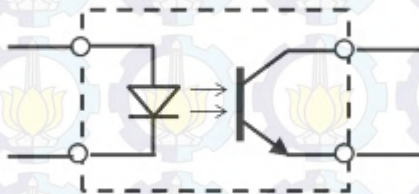
Arus *eddy* merupakan arus listrik yang timbul jika suatu piringan logam berada di sekitar medan magnet yang garis-garis gayanya sedang berubah-ubah. Arus *eddy* ini mempunyai medan magnet yang arahnya berlawanan dengan arah gerak piringan logam. Akibatnya laju piringan logam akan tertahan akibat dari adanya arus *eddy* ini.

Rem elektromagnetik biasa diletakkan dekat dengan bagian yang bergerak. Rem ini bekerja pada kondisi yang dingin dan memenuhi persyaratan energi pengereman kecepatan tinggi karena tanpa adanya gesekan. Selain pada kendaraan listrik, aplikasi rem elektromagnetik juga dapat ditemukan pada sistem pengereman kereta api.

2.3 Rangkaian *Optocoupler*

Rangkaian *optocoupler* digunakan untuk memisahkan perangkat Arduino dengan *plant* sehingga perangkat Arduino dapat terhindar dari kerusakan akibat arus berlebih dari *power supply* jika terjadi kecelakaan.

Rangkaian isolator yang digunakan terdiri dari *optocoupler* PIC817 dengan rangkaian yang dapat dilihat pada Gambar 2.5

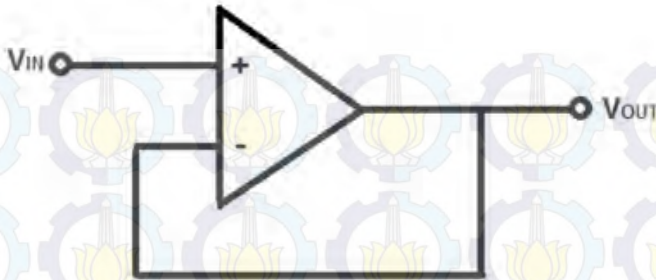


Gambar 2.5 Rangkaian *Optocoupler* [6]

2.3.1 Rangkaian *Voltage Follower*

Rangkaian *voltage follower* berguna agar tidak terjadi pembagian arus yang berakibat pada *drop* tegangan. *Drop* tegangan akan menyebabkan kesalahan pada pembacaan sinyal.

Rangkaian *voltage follower* yang digunakan terdiri dari sebuah *op-amp* dengan rangkaian yang dapat dilihat pada Gambar 2.6



Gambar 2.6 Rangkaian *Voltage Follower* [7]

2.4 Identifikasi Sistem [8]

Identifikasi berarti penentuan model dari sistem dinamis melalui pengukuran *input-output*. Pengetahuan tentang model diperlukan untuk mendesain dan mengimplementasikan sebuah sistem pengaturan. Terdapat 2 tipe dari model dinamis yaitu model *non-parametris* (respon frekuensi, respon *step*) dan model parametris (fungsi *transfer*, persamaan beda).

Identifikasi sistem adalah pendekatan eksperimental untuk menentukan model dinamis dari sebuah sistem. Terdapat 4 langkah dalam identifikasi sistem: akuisisi data *input-output*, penentuan atau estimasi dari struktur model, estimasi parameter model, validasi dari model yang telah teridentifikasi.

Pada akuisisi data *input-output*, sinyal *input* yang dipilih harus memiliki spektrum frekuensi yang kaya. Umumnya, sinyal *input* dipilih berupa sinyal PRBS.

Masalah sering yang timbul saat melakukan identifikasi sistem adalah menentukan derajat dari polinomial (*numerator* dan *denominator*) dari fungsi alih yang merepresentasikan model *plant*. Prosedur *trial and error* banyak digunakan dalam penentuan kompleksitas model.

Salah satu elemen kunci untuk mengimplementasikan identifikasi sistem adalah algoritma adaptasi parameter yang mengubah parameter dari model prediksi dari data yang didapatkan pada tiap waktu *sampling*.

Pada tahap validasi, model yang telah didapatkan diberikan *input* yang sama dengan yang diberikan ke *plant*. *Output* dari model dan *plant*.

2.5 Fuzzy Logic [9]

Teori logika konvensional, secara matematis, menerapkan batas yang jelas antar kategori. Sebagai contoh, apabila didefinisikan bahwa orang tinggi adalah orang yang memiliki tinggi di atas 170 cm. Maka orang yang memiliki tinggi 169,9 cm termasuk kategori orang pendek. Hal seperti ini secara intuitif tidak masuk akal menurut pemikiran manusia karena walaupun memiliki selisih yang sangat sedikit, orang yang memiliki tinggi 169,9 cm dikategorikan orang pendek dibandingkan dengan orang yang memiliki tinggi 170,1 cm.

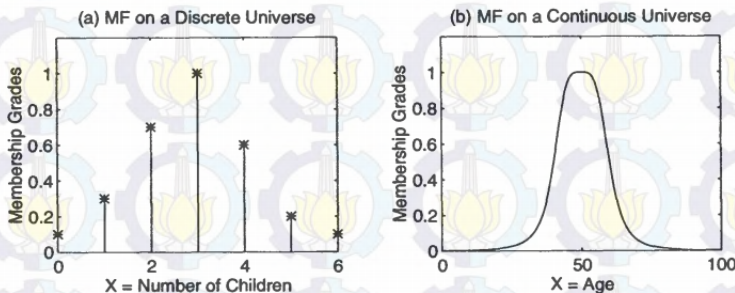
Dalam logika *fuzzy*, tidak terdapat batas yang jelas, melainkan terdapat transisi antara *set* yang satu dengan *set* yang lainnya. Transisi ini dikarakterisasi oleh fungsi keanggotaan yang memberikan sebuah *fuzzy set* fleksibilitas dalam memodelkan ekspresi linguistik.

2.5.1 Fuzzy Set dan Membership Function

Misalkan X adalah semesta sebuah obyek dan x adalah elemen dari X , maka sebuah *fuzzy set* A di dalam X didefinisikan sebagai:

$$A = \{(x, \mu_A(x)) \mid x \in X\} \quad (2.4)$$

Di mana $\mu_A(x)$ adalah *membership function* (MF) dari *fuzzy set* A . MF menetapkan tiap elemen X dalam sebuah nilai keanggotaan antara 0 sampai 1. Contoh sebuah MF ditunjukkan pada Gambar 2.7



Gambar 2.7 Contoh Fungsi Keanggotaan

2.5.2 Fuzzy If-Then Rule

Sebuah *fuzzy if-then rule* memiliki bentuk:

JIKA x adalah A, MAKA y adalah B

Di mana A dan B adalah nilai linguistik yang didefinisikan oleh *fuzzy set* dalam semesta X dan Y.

2.5.3 Sugeno Fuzzy Model

Sugeno Fuzzy Model yang digunakan dalam tugas akhir ini adalah salah satu model sistem penalaran *fuzzy*. *Fuzzy rule* pada *Sugeno fuzzy model* memiliki bentuk umum

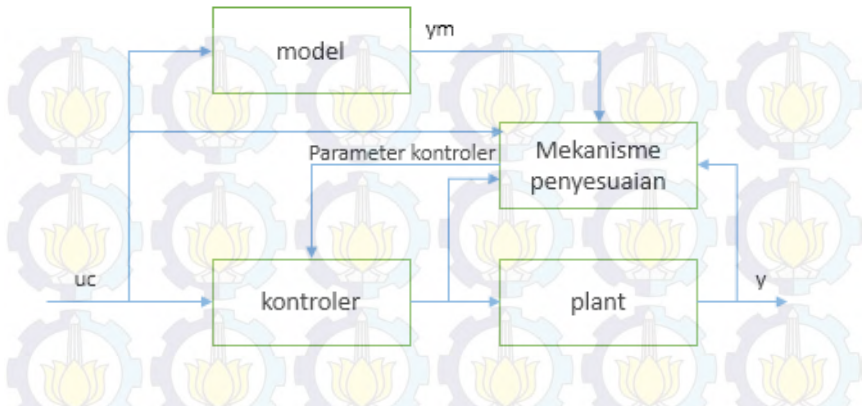
Jika x adalah A dan y adalah B, maka $z = f(x, y)$

Biasanya, $f(x, y)$ berbentuk polinomial dari variabel *input* x dan y . Ketika $f(x, y)$ merupakan polinomial orde 1, maka sistem penalaran *fuzzy* ini disebut *first-order sugeno fuzzy model*. Ketika f adalah sebuah konstanta, maka disebut *zero-order sugeno fuzzy model*.

2.6 Model Reference Adaptive System [10]

Model Reference Adaptive System (MRAS) adalah kontroler adaptif yang penting. Dalam MRAS performa sistem yang diinginkan dinyatakan dalam suatu model referensi. Diagram blok dari sistem ditunjukkan pada Gambar 2.8. Sistem memiliki *loop* umpan balik yang terdiri dari proses dan kontroler serta *loop* umpan balik yang lain yang mengubah parameter kontroler. Parameter diubah berdasarkan umpan balik dari *error*, yang merupakan selisih antara *output* dari sistem dan *output* dari model referensi. *Loop* umpan balik pertama disebut *inner loop* dan *loop* kedua disebut *outer loop*. Mekanisme penyesuaian parameter dalam MRAS dapat diperoleh dengan dua cara: dengan menggunakan metode gradien atau dengan menerapkan teori stabilitas.

Dalam MRAS perilaku yang diinginkan dari sistem ditentukan oleh sebuah model dan parameter kontroler disesuaikan berdasarkan *error* yang merupakan selisih antara *output* dari sistem *loop* tertutup dan model tersebut.



Gambar 2.8 Diagram Blok *Model Reference Adaptive System*

2.6.1 MRAS berbasis *Output Feedback*

Di sini akan secara umum dirumuskan sebuah MRAS untuk menyesuaikan parameter kontroler berbasis pada *output feedback*. Sistem dianggap berupa sistem SISO dan linier.

2.6.1.1 Menentukan Struktur Kontroler

Anggap proses dideskripsikan dengan *continuous-time model*:

$$Ay(t) = b_0 Bu(t) \quad (2.5)$$

Sebuah kontroler linier umum dapat ditulis sebagai:

$$Ru(t) = -Sy(t) + Tu_c(t) \quad (2.6)$$

Di mana u_c adalah sinyal referensi. Jika dipilih $R = R_1 B$. Persamaan sistem *closed-loop* menjadi:

$$(AR_1 + b_0 S)y = b_0 Tu_c \quad (2.7)$$

Jika dipilih $T = t_0 A_0$ di mana A_0 adalah *stable monic polynomial* dan R_1 dan S memenuhi:

$$AR_1 + b_0S = A_0A_m \quad (2.8)$$

Maka dimungkinkan untuk mendapatkan sistem yang memiliki respon mengikuti model:

$$A_my_m(t) = b_0t_0u_c(t) \quad (2.9)$$

2.6.1.2 Error Model

Dari persamaan sebelum sebelumnya dapat diperoleh

$$A_0A_my = AR_1y + b_0Sy = R_1b_0Bu + b_0Sy \quad (2.10)$$

Misal:

$$e = y - y_m \quad (2.11)$$

Dari Persamaan 2.11 dan 2.10 didapatkan:

$$A_0A_me = A_0A_m(y - y_m) = b_0(Ru + Sy - Tu_c) \quad (2.12)$$

Atau

$$e = \frac{b_0}{A_0A_m}(Ru + Sy - Tu_c) \quad (2.13)$$

Berikutnya diperkenalkan *filtered error*:

$$e_f = \frac{Q}{P}e = \frac{Q}{P}(y - y_m) \quad (2.14)$$

Di mana Q adalah sebuah polinomial dengan derajat yang tidak lebih dari derajat polinomial A_0A_m . Berikutnya *filtered error* dapat ditulis sebagai

$$e_f = \frac{Q}{P}\left(\frac{S}{P}u + \frac{S}{P}y - \frac{T}{P}u_c\right) \quad (2.15)$$

Misalkan $P = P_1 P_2$, di mana P_2 adalah *stable monic polynomial* yang memiliki derajat yang sama dengan R . Tulis kembali R/P menjadi:

$$\frac{R}{P} = \frac{R - P_2 + P_2}{P_1 P_2} = \frac{1}{P_1} + \frac{R - P_2}{P} \quad (2.16)$$

Sehingga *filtered error* menjadi:

$$e_f = \frac{b_0 Q}{A_0 A_m} \left(\frac{1}{P_1} u + \frac{R - P_2}{P} u + \frac{S}{P} y - \frac{T}{P} u_c \right) \quad (2.17)$$

Misalkan k , l , dan m adalah derajat dari polinomial R , S , dan T . Dan misalkan sebuah *true controller parameter vector*:

$$\theta^0 = (r'_1 \dots r'_k s_0 \dots s_l t_0 \dots t_m)^T \quad (2.18)$$

Di mana r'_i adalah koefisien dari polinomial $R - P_2$. Kemudian dimisalkan juga vektor *filtered input, output*, dan sinyal referensi:

$$\varphi^T = \left(\frac{p^{k-1}}{P(p)} u \dots \frac{1}{P(p)} u \quad \frac{p^l}{P(p)} y \dots \frac{1}{P(p)} y \quad - \frac{p^m}{P(p)} u_c \dots - \frac{1}{P(p)} u_c \right) \quad (2.19)$$

Sehingga *filtered error* dapat ditulis sebagai:

$$e_f = \frac{b_0 Q}{A_m A_m} \left(\frac{1}{P_1} u + \varphi^T \theta^0 \right) \quad (2.20)$$

Untuk mendapatkan *error model*, diperkenalkan sebuah parameterisasi dari kontroler. Dalam sebuah kasus di mana parameter-parameter diketahui, aturan kontrol dapat dinyatakan dengan:

$$u = -P_1(\varphi^T \theta^0) = -P_1((\theta^0)^T \varphi) = -(\theta^0)^T (P_1 \varphi) \quad (2.21)$$

Di mana P_1 adalah sebuah polinomial. Misalkan θ melambangkan *adjustable parameter controller*, aturan *feedback*:

$$u = -(\theta)^T (P_1 \varphi) \quad (2.22)$$

Akan memberikan model *error* yang diinginkan. Kemudian, *filtered error* dapat ditulis menjadi:

$$e_f = \frac{b_0 Q}{A_0 A_m} \left(\varphi^T \theta^0 - \varphi^T \theta - \frac{1}{P_1} \theta^T (P_1 \varphi) + \varphi^T \theta \right) \quad (2.23)$$

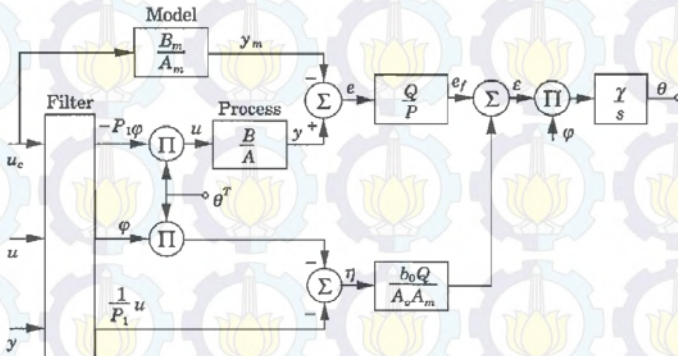
Definisikan sebuah sinyal η dan ε :

$$\begin{aligned} \eta &= \frac{1}{P_1} \theta^T (P_1 \varphi) - \varphi^T \theta = - \left(\frac{1}{P_1} u + \varphi^T \theta \right) \\ \varepsilon &= e_f + \frac{b_0 Q}{A_0 A_m} \eta = \frac{b_0 Q}{A_0 A_m} \varphi^T (\theta^0 - \theta) \end{aligned} \quad (2.24)$$

Sinyal ε disebut dengan *augmented error*, dan η disebut dengan *error augmentation*. Dari Persamaan 2.25 diperoleh bahwa nilai *high frequency gain* b_0 harus diketahui. Jika b_0 tidak diketahui, maka dapat diprediksi dengan persamaan berikut:

$$\frac{d\hat{b}_0}{dt} = \gamma (\varphi_f^T \theta + u_f) (e_f - \hat{b}_0 (\varphi_f^T \theta + u_f)) \quad (2.25)$$

Blok diagram dari *model reference adaptive system* dapat dilihat pada Gambar 2.9.

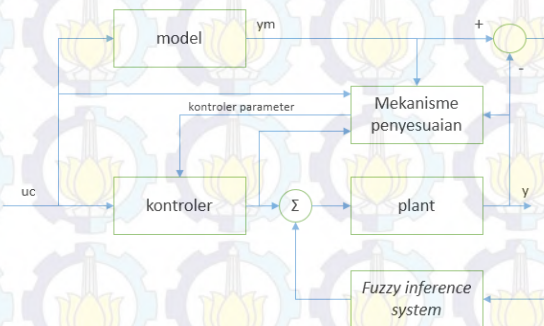


Gambar 2.9 Diagram Blok *Model-Reference Adaptive System*

Prosedur perancangan MRAC mencakup pemilihan beberapa parameter berikut: fungsi *transfer* model B_m/A_m , polinomial A_0 , derajat polinomial R , S , dan T , serta polinomial P_1 , P_2 , dan Q .

2.7 Fuzzy Modified MRAC

Gambar 2.10 menunjukkan blok diagram dari sistem *Fuzzy Modified MRAC* yang akan dirancang pada tugas akhir kali ini. Sistem ini memiliki *loop* tambahan yang mengumpan balik *error output plant* dan model melalui *fuzzy inference system*. Keluaran *fuzzy inference system* ditambahkan ke sinyal kontrol agar waktu konvergensi *output plant* dan model menjadi lebih cepat.



Gambar 2.10 Blok Diagram *Fuzzy Modified MRAC* [11]

BAB 3

PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem

Sistem yang saya buat merupakan sistem pengaturan motor BLDC. Motor BLDC yang digunakan adalah motor yang biasa digunakan dalam mesin *air conditioner*. Motor ini dikendalikan dengan *input* berupa tegangan dengan rentang 0 sampai 5 Volt. Informasi kecepatan diperoleh dari keluaran *driver* yang ada dalam *case* motor berupa sinyal kotak yang frekuensinya berubah tergantung kecepatan putar rotor. Karena itu, tidak diperlukan tambahan sensor kecepatan. Untuk selanjutnya *plant* motor BLDC yang kami gunakan akan kami beri nama P-1 .

Pada sistem ini, dipasang sebuah rem magnetik yang terdiri dari cakram alumunium dan kumparan yang bekerja menurut prinsip *eddy current*. Rem magnetik ini berguna untuk mengubah model matematik *plant* sehingga dapat diketahui performa kontroler terhadap perubahan model matematik *plant*.

Sebagai kontroler, digunakan laptop yang telah ter-*install* MATLAB.

3.2 Perancangan Perangkat Keras

Perancangan perangkat keras terdiri dari perancangan mekanik dan perancangan elektrik. Perancangan mekanik berisi pembuatan maket *plant* yang terdiri dari motor, rem magnetik dll. Sedangkan pada perancangan elektrik, berisi pengkabelan Arduino, rangkaian isolasi, dan rangkaian *voltage follower* dll.

3.2.1 Perancangan Mekanik

Plant P-1 terdiri dari komponen utama berupa motor dan rem magnetik. *Shaft* motor diperpanjang dengan silinder besi yang dihubungkan dengan motor menggunakan sebuah kopling. Perpanjangan *shaft* digunakan untuk pemasangan cakram alumunium yang merupakan salah satu komponen rem elektromagnetik. Motor didudukkan pada suatu maket yang terbuat dari akrilik yang juga digunakan sebagai dudukan dari kumparan rem.

3.2.1.1 Motor BLDC

Motor yang digunakan pada tugas akhir ini merupakan motor yang diambil dari sebuah *air conditioner* buatan Daikin. Motor ini memiliki *built-in driver* dengan 5 buah kabel *input-output*. Masing-masing kabel memiliki warna isolasi berbeda dengan fungsi yang dapat dilihat pada Tabel 3.1. Motor ini memiliki 4 buah pasangan kutub.

Tabel 3.1 Daftar Warna Kabel *Input-Output Driver* Motor dan Fungsinya

Warna	Keterangan
Jingga	Kabel <i>input</i> sinyal kontrol. <i>Input</i> kabel ini merupakan tegangan dengan rentang 0-5 Volt. Kecepatan putar motor akan berubah sesuai dengan tegangan yang diberikan pada kabel ini.
Putih	Kabel <i>output</i> sinyal informasi kecepatan motor. Kabel ini men- <i>generate</i> sinyal kotak dengan frekuensi sesuai dengan kecepatan putar motor. Hubungan antara frekuensi dan kecepatan motor adalah $\omega = 15 \times f$ Di mana ω adalah kecepatan putar motor dalam rpm dan f adalah frekuensi sinyal dalam Hertz.
Merah	Kabel <i>power</i> yang men- <i>supply</i> daya ke motor.
Biru	Kabel <i>common</i> dari <i>driver</i> dan motor.
Coklat	Kabel <i>power</i> yang men- <i>supply</i> daya ke <i>driver</i> .

3.2.1.2 Rem Elektromagnetik

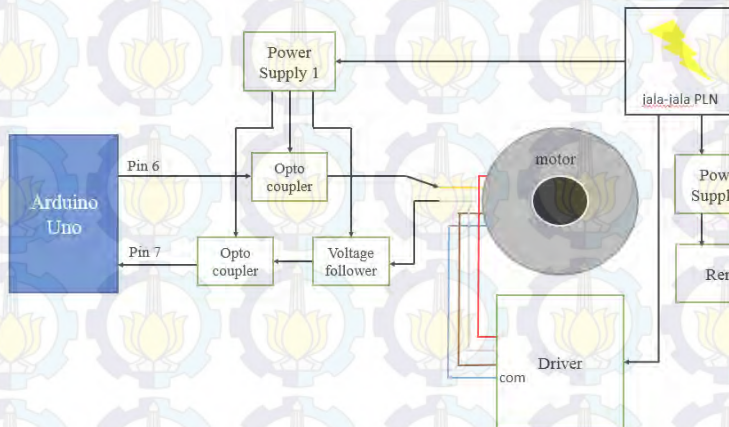
Rem elektromagnetik yang kami buat terdiri dari 1 cakram alumunium dan 4 pasang kumparan. Sebuah baut besi digunakan sebagai inti kumparan. Inti dililit dengan kawat tembaga 0,4 mm. Lilitan ini terdiri dari beberapa lapisan yang tiap lapisannya dipisahkan dengan kertas isolasi. Tiap kumparan memiliki panjang 6 cm dan diameter 2 cm dengan lilitan sebanyak 400 lilitan. Untuk cakram, dibuat dari alumunium dengan diameter 15 cm dan tebal 0,5 cm.

Kumparan-kumparan dihubungkan secara seri dan dialiri dengan arus listrik DC dengan besar yang dapat diatur sesuai dengan besar tegangan dari *power supply*.

3.2.2 Perancangan Elektrik

Konfigurasi perkabelan komponen elektrik *plant* yang dapat dilihat pada Gambar 3.1. Pin 6 Arduino memberikan *output* berupa PWM

dengan amplitudo 5 V. Sinyal PWM tersebut dilewatkan ke sebuah *optocoupler* dan diumpangkan ke kabel jingga motor. Informasi kecepatan didapat dari frekuensi sinyal yang keluar dari kabel putih. Sebelum masuk Arduino, sinyal tersebut terlebih dahulu diumpangkan ke sebuah *voltage follower*. Tujuannya adalah agar tidak terjadi *drop* tegangan. Sinyal keluaran rangkaian *voltage follower* kemudian diumpangkan ke Arduino dengan terlebih dahulu dilewatkan rangkaian isolasi berupa *optocoupler*.

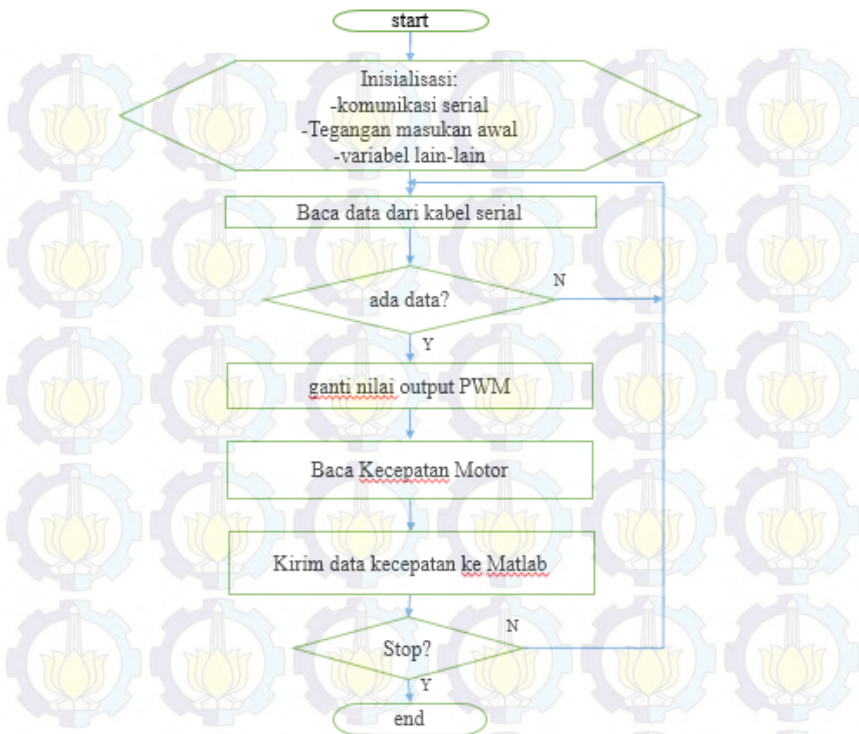


Gambar 3.1 Pengkabelan Sistem Elektrik *Plant*

3.3 Perancangan Perangkat Lunak

3.3.1 Perangkat Lunak Arduino

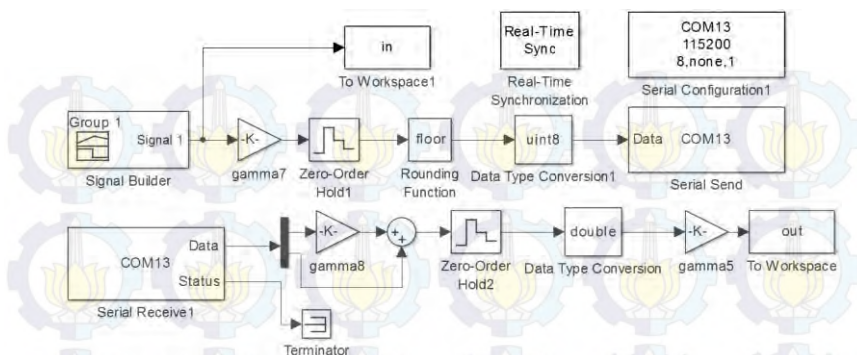
Pada tugas akhir ini, Arduino digunakan sebagai *interface* yang menghubungkan *plant* dengan laptop melalui komunikasi *serial*. Arduino menerima data sinyal kontrol yang dikirim oleh perangkat lunak Matlab dan mengeluarkan sinyal PWM sesuai dengan data tersebut untuk kemudian menjadi masukan dari *plant*. Kemudian, Arduino menghitung kecepatan putar motor dan mengirimkan nilainya ke Matlab. Matlab mengolah data kecepatan putar motor untuk kemudian mengirimkan kembali sinyal kontrol hasil pengolahan kecepatan putar motor ke Arduino dan begitu seterusnya. Diagram alir dari *program* yang dipasang pada Arduino dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Alir Program Arduino

3.4 Perangkat Lunak Matlab

Perangkat lunak Matlab menerima data kecepatan motor dari Arduino dan kemudian mengirimkan sinyal kontrol ke Arduino untuk diteruskan ke *plant*. Pada tahap identifikasi, sinyal kontrol dari Matlab di-*generate* dari blok *signal builder*. Sedangkan pada tahap simulasi dan implementasi, sinyal kontrol adalah hasil pemrosesan kontroler berdasarkan masukan kecepatan. Untuk tahap identifikasi, gambar diagram blok Simulink-nya dapat dilihat pada Gambar 3.3



Gambar 3.3 Diagram Blok Simulink untuk Identifikasi

Blok *signal builder* sinyal PRBS dengan batas bawah 0,7 dan batas atas 0,9 dengan maksud bahwa nantinya sinyal kontrol maksimum yang dikeluarkan kontroler adalah 1. Keluaran blok *signal builder* dikalikan dengan 255 untuk penskalaan pada *output* PWM Arduino. Data tersebut kemudian dilewatkan ke blok *zero-order hold* yang berguna sebagai *buffer*. Data kemudian dibulatkan menggunakan blok *rounding function* dan diubah tipe datanya menjadi data *byte* menggunakan blok *data type conversion*. Setelah itu, data dikirimkan ke Arduino menggunakan blok *Serial send*.

Pada sisi penerimaan, data kecepatan motor diterima oleh Matlab dan dikeluarkan oleh blok *Serial receive*. Data yang dikirimkan oleh Arduino merupakan data *uint16* yang dikirimkan sepotong-sepotong sebesar 8 bit. Data pertama dikalikan dengan 256 dan ditambahkan dengan data kedua untuk mendapatkan kembali data *uint16* kecepatan motor. Data kemudian di-*buffer* dengan blok *zero-order hold* dan dikonversi ke bentuk tipe data *double*. Data tersebut kemudian dibagi dengan nilai kecepatan kerja maksimum untuk mendapatkan hasil penskalaan 0-1 dari nilai kecepatan.

3.5 Identifikasi dan Pemodelan Sistem

Proses identifikasi dilakukan dengan menggunakan *System Identification Toolbox* yang terdapat pada Matlab. Dengan aplikasi ini, pengguna dapat dengan mudah mendapatkan model matematika *plant* dengan memasukkan data *input-output*, melakukan *preprocessing*, dan memilih bentuk model yang diinginkan. Pada tugas akhir kali ini, saya

ingin mendapatkan model matematika *plant* dalam bentuk fungsi *transfer* kontinu.

Fungsi *transfer* hasil identifikasi dibatasi hanya berupa fungsi *transfer* orde dua tanpa *zero*. Hasil identifikasi pemodelan *plant* menggunakan *System Identification Toolbox* Matlab dapat dilihat pada Tabel 3.2.

Tabel 3.2 Fungsi *Transfer Plant* Hasil Identifikasi

Keadaan beban	Tegangan input beban	Fungsi <i>Transfer</i>	RMSE %
Minimal	16 Volt	$\frac{98,578}{s^2 + 25,3125s + 98,578}$	5,31
Nominal	20 Volt	$\frac{122,2386}{s^2 + 28,9879s + 134,5795}$	4,89
Maksimal	24 Volt	$\frac{232,7394}{s^2 + 49,9617s + 273,8755}$	3,41

3.6 Perancangan *Fuzzy-MRAC*

3.6.1 Perancangan MRAC

Pertama, fungsi *transfer* model dipilih berupa fungsi *transfer* Butterworth orde 2 yang memiliki bentuk:

$$\frac{1}{s^2 + 1,414s + 1} \quad (3.1)$$

Fungsi *transfer* ini dipilih karena fungsi *transfer* ini mengikuti indeks performansi minimal ITAE.

Untuk pemilihan polinomial A_0 , dipilih $A_0 = s + 2$. Sedangkan untuk pemilihan polinomial P_1 , P_2 , dan Q , dipilih $P_1 = A_m$, $P_2 = A_0$, dan $Q = A_0 A_m$. Pemilihan tersebut adalah pemilihan yang populer karena menyebabkan $Q/P = 1$, sehingga kontroler menjadi lebih sederhana. Sedangkan untuk memilih derajat polinomial R , S , dan T , dipilih derajat k , l dan m bernilai 1.

Untuk mekanisme penyesuaian, dipakai mekanisme PI dengan Persamaan 3.2.

$$\theta = \gamma_1 \varphi \varepsilon + \gamma_2 \int \varphi \varepsilon dt \quad (3.2)$$

3.6.1.1 Vektor φ dan $P_1\varphi$

Dari sub-bab sebelumnya didapatkan nilai orde polinomial RST $k = 1$, $l = 1$, dan $m = 1$. Sedangkan untuk polinomial $P = P_1P_2$, di mana telah ditentukan bahwa:

$$P_1 = A_m = s^2 + 1.414s + 1 \quad (3.3)$$

$$P_2 = A_0 = s + 2 \quad (3.4)$$

Sehingga, dari Persamaan 2.21

$$\varphi = \frac{1}{(s^2 + 1.414s + 1)(s + 2)} \begin{bmatrix} (1)u \\ (s)y \\ (1)y \\ (s)u_c \\ (1)u_c \end{bmatrix} \quad (3.5)$$

Dan

$$P_1\varphi = \frac{1}{(s + 2)} \begin{bmatrix} (1)u \\ (s)y \\ (1)y \\ (s)u_c \\ (1)u_c \end{bmatrix} \quad (3.6)$$

3.6.2 Perancangan Kontroler Fuzzy

Kontroler *fuzzy* digunakan untuk memperbaiki sinyal *kontrol* agar kecepatan konvergensi *output plant* dan *output model* menjadi lebih cepat. Kontroler *fuzzy* yang dirancang merupakan model *fuzzy* Takagi-Sugeno orde nol.

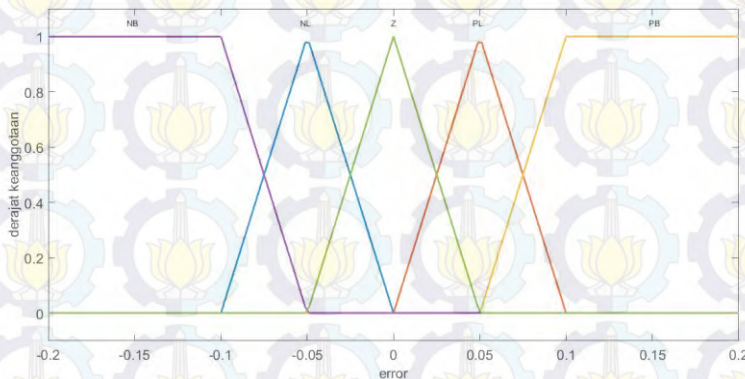
Kontroler *fuzzy* mendapatkan masukan *error* antara *output plant* dan *output model*. Nilai *error* tersebut kemudian dikategorikan menjadi 3 variabel linguistik negatif, *zero*, dan positif.

3.6.2.1 Fungsi Keanggotaan

Ketiga variabel linguistik tersebut kemudian didefinisikan dalam suatu fungsi keanggotaan sebagai berikut

$$\begin{aligned}\mu_{NB}(\text{error}) &= \text{trapesium}(\text{error}; -2; -2; -0,1; -0,05) \\ \mu_{NL}(\text{error}) &= \text{triangle}(\text{error}; -0,1; -0,05; 0) \\ \mu_Z(\text{error}) &= \text{triangle}(\text{error}; 0,05; 0; 0,05) \\ \mu_{PL}(\text{error}) &= \text{triangle}(\text{error}; 0; 0,05; 0,1) \\ \mu_{PB}(\text{error}) &= \text{triangle}(\text{error}; 0,05; 0,1; 2; 2)\end{aligned}\quad (3.6)$$

Atau dapat dilihat pada Gambar 3.4.



Gambar 3.4 Fungsi Keanggotaan *Input* Kontroler Fuzzy

3.6.2.2 Basis Aturan

Setelah menetapkan fungsi keanggotaan dari *input*, langkah selanjutnya adalah membuat basis aturan. Basis aturan yang dibuat adalah sebagai berikut.

Jika	<i>error</i> negatif <i>big</i> ,	maka	<i>output</i> = 0,5
Jika	<i>error</i> negatif <i>low</i> ,	maka	<i>output</i> = 0,25
Jika	<i>error</i> zero,	maka	<i>output</i> = 0
Jika	<i>error</i> positif <i>low</i> ,	maka	<i>output</i> = -0,25
Jika	<i>error</i> positif <i>big</i> ,	maka	<i>output</i> = -0,5

Output keseluruhan dari kontroler *fuzzy* ini kemudian dihitung menggunakan operasi *weighted sum*, yaitu

$$y = \mu_{NB}(e)(0,5) + \mu_{NL}(e)(0,25) + \mu_Z(e)(0) + \mu_{BL}(e)(-0,25) + \mu_{PB}(e)(-0,5) \quad (3.6)$$

Di mana y adalah *output* kontroler *fuzzy* dan e adalah *error* dari *output plant* dan model.

BAB 4

PENGUJIAN DAN ANALISA

4.1 Gambaran Umum Pengujian Sistem

Pada tahapan ini, terdapat beberapa hal yang diuji. Diantaranya adalah pengujian sensor kecepatan motor, pengujian *open loop* kecepatan motor, pengujian kontroler dengan simulasi dan pengujian kontroler pada *plant real*.

Pengujian sensor kecepatan motor dilakukan dengan menghitung nilai frekuensi sinyal *output driver* motor, kemudian membandingkannya dengan hasil bacaan dari *tachogenerator*. Hal ini berguna untuk mengetahui hubungan frekuensi sinyal dengan kecepatan motor.

Pengujian *open loop* kecepatan motor dilakukan dengan memberikan *input* tegangan pada motor dari 0 Volt sampai 5 Volt secara bertahap. Kecepatan motor kemudian dibaca pada tiap tahapan sehingga didapat grafik *input-output plant*.

Pengujian simulasi dilakukan dengan mensimulasikan kontroler yang telah dirancang di perangkat lunak Matlab. Simulasi ini menggunakan fungsi *transfer plant* hasil identifikasi yang telah diterangkan di Bab 3.

Pengujian terakhir adalah implementasi kontroler ke *plant* sebenarnya.

4.2 Pengujian Sensor Kecepatan

Kecepatan motor diukur dengan menghitung frekuensi dari sinyal kotak oleh *driver* bawaan motor. Kecepatan motor dihitung dengan hubungan $\omega = 15 \times f$. Hasil pembacaan Arduino kemudian dibandingkan dengan hasil pembacaan dengan *tachogenerator*. Hasil pengujian sensor kecepatan ini dapat dilihat pada Tabel 4.1.

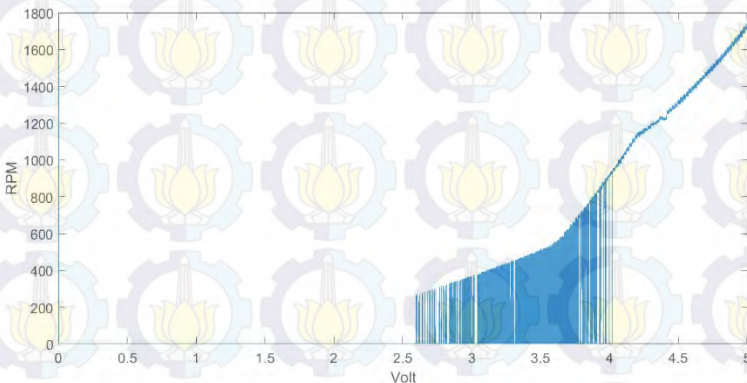
Dari hasil pengujian diketahui bahwa hasil pembacaan yang didapatkan oleh Arduino tidak berbeda jauh dengan hasil pembacaan dari *tachogenerator* dengan *error* berkisar antara 1,11 - 1,84 %. Sehingga dapat disimpulkan bahwa metode pengukuran kecepatan motor dengan menghitung frekuensi pulsa dari keluaran *driver* motor sudah benar.

Tabel 4.1 Hasil Pengujian Sensor Kecepatan

Hasil Pembacaan		% Error
Arduino	Tachogenerator	
972.36	984.6	1,25
1022.1	1035.41	1,30
1059	1070	1,11
1201.8	1218.04	1,35
1292,8	1310,42	1,36
1363.4	1382.67	1,41
1429,5	1448,82	1,35
1472,5	1491,69	1,3
1703.6	1726	1,31
2253	2290,45	1,67
2397	2441,21	1,84

4.3 Pengujian *Open Loop* Kecepatan Motor

Pengujian ini dilakukan untuk mengetahui grafik hubungan *input-output plant*. Pengujian dilakukan pada saat beban nominal dengan memberikan sinyal tangga pada *input plant* dan diukur kecepatannya. Hasil dari pengujian dapat dilihat pada Gambar 4.1.

**Gambar 4.1** Hubungan *Input-Output Plant*

Dari gambar tersebut, terlihat bahwa pada saat kecepatan dibawah 1000 rpm, kecepatan motor tidak terukur dengan baik. Hal ini

menunjukkan kekurangan dari sensor kecepatan motor sehingga rentang kerja dari *plant* ini tidak boleh dibawah 1000 rpm.

Pada kecepatan di atas 1000 rpm, terlihat bahwa hubungan *input-output plant* relatif linier dengan batas maksimum kecepatan motor berkisar pada 1700 rpm. Maka, pada tugas akhir ini, dipilih rentang kerja motor antara 1000-1600 rpm. Dengan *input* tegangan antara 4-5 Volt

4.4 Simulasi Sistem

4.4.1 Blok Diagram Simulink

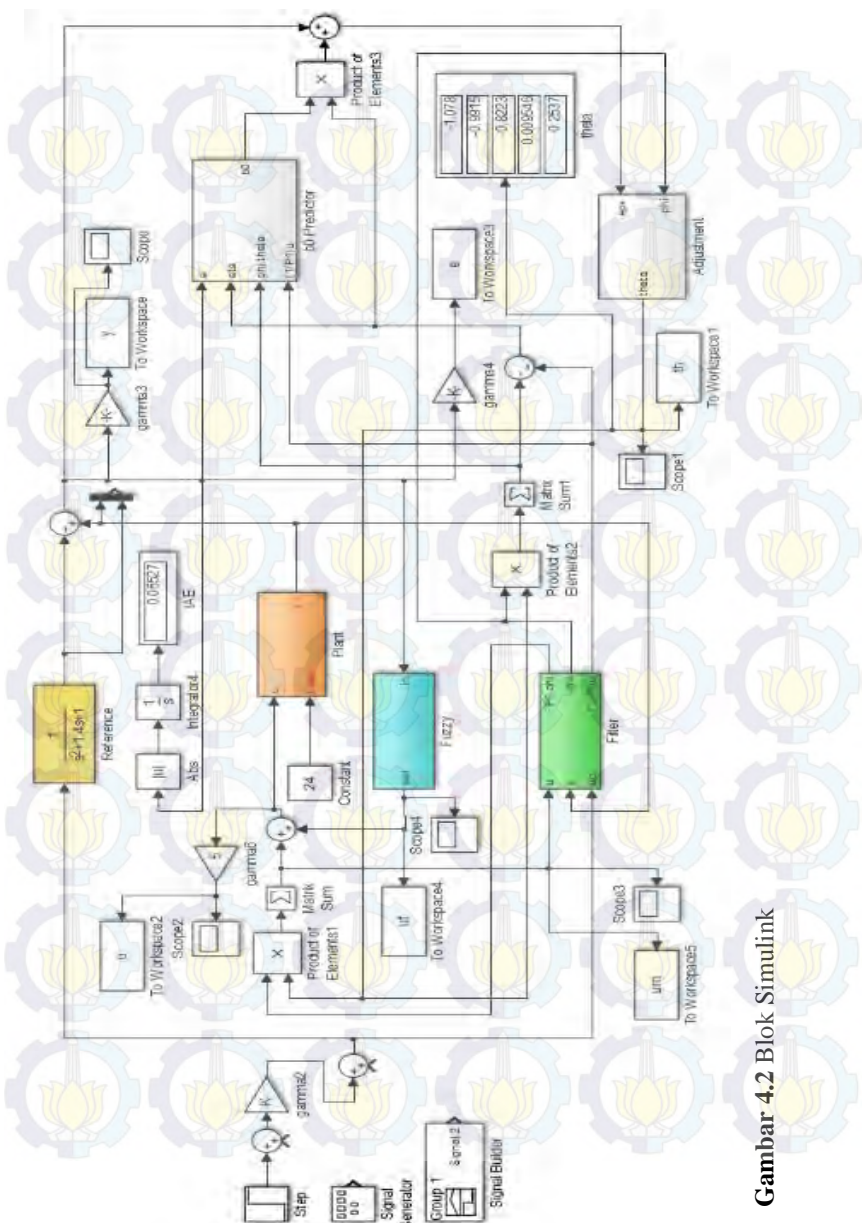
Hasil rancangan kontroler yang terdapat pada Bab 3 diimplementasikan dalam bentuk blok diagram simulink untuk disimulasikan. Blok diagram simulink sistem dapat dilihat pada Gambar 4.2

Pada blok diagram tersebut, sinyal referensi u_c menjadi masukan dari model referensi dan blok *filter*. Blok *filter* menerima masukan dari u_c , sinyal kontrol u , dan sinyal keluaran *plant* y untuk kemudian memberikan keluaran vektor $-P_1\varphi$, φ , dan sinyal kontrol terfilter $1/p_1 u$.

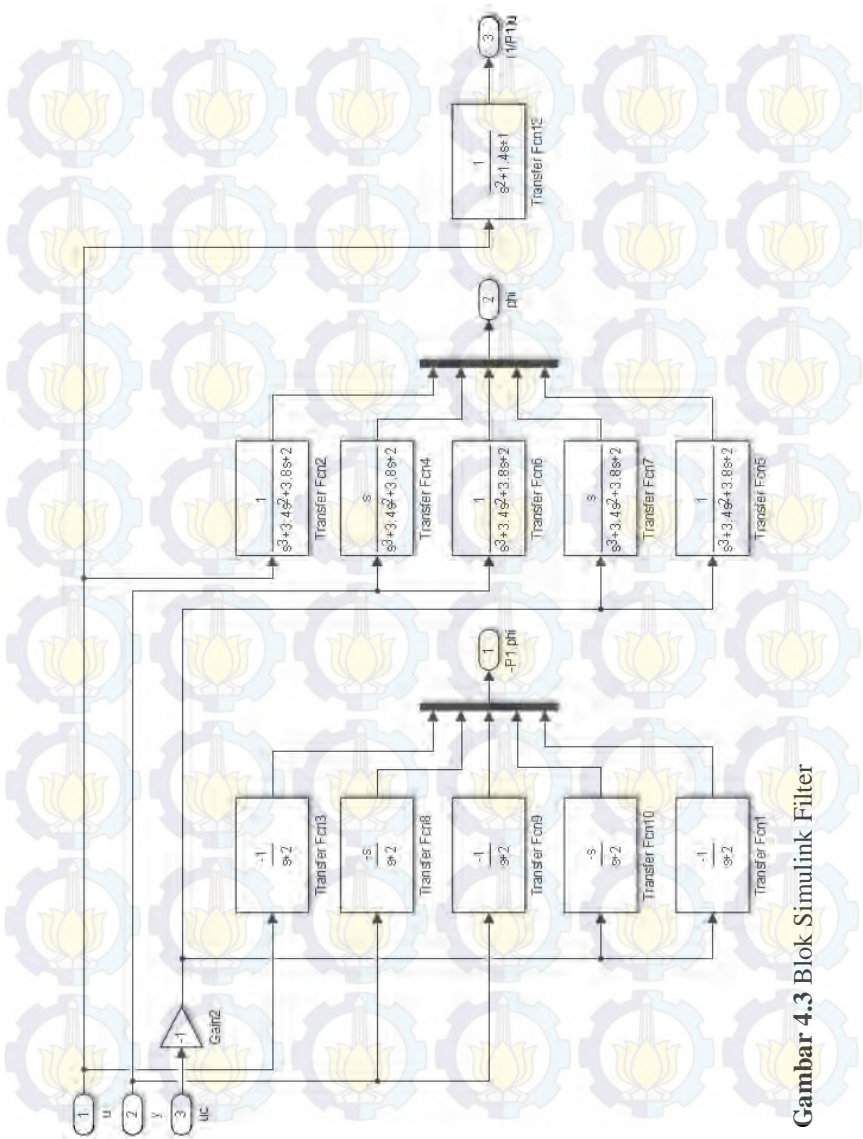
Sinyal kontrol u dibentuk dari persamaan $u = -P_1\varphi(\theta)$ yang merupakan keluaran blok *filter* dan blok *adjustment*. Sementara b_0 dibentuk dari masukan e , $\varphi_f^T\theta$, dan u_f dengan hubungan sesuai dengan Persamaan 2.27. Kemudian pada blok *adjustment*, keluaran parameter kontroler θ dibentuk dari masukan *augmented error* ε , dan φ .

Gambar 4.3 menunjukkan bagian dalam dari blok *filter*. Susunan dari blok tersebut berdasarkan pada Persamaan 2.21 dengan $P_1 = s^2 + 1,414s + 1$ dan $P_2 = s + 2$ sehingga didapatkan $P = s^3 + 3,4s^2 + 3,8s + 2$.

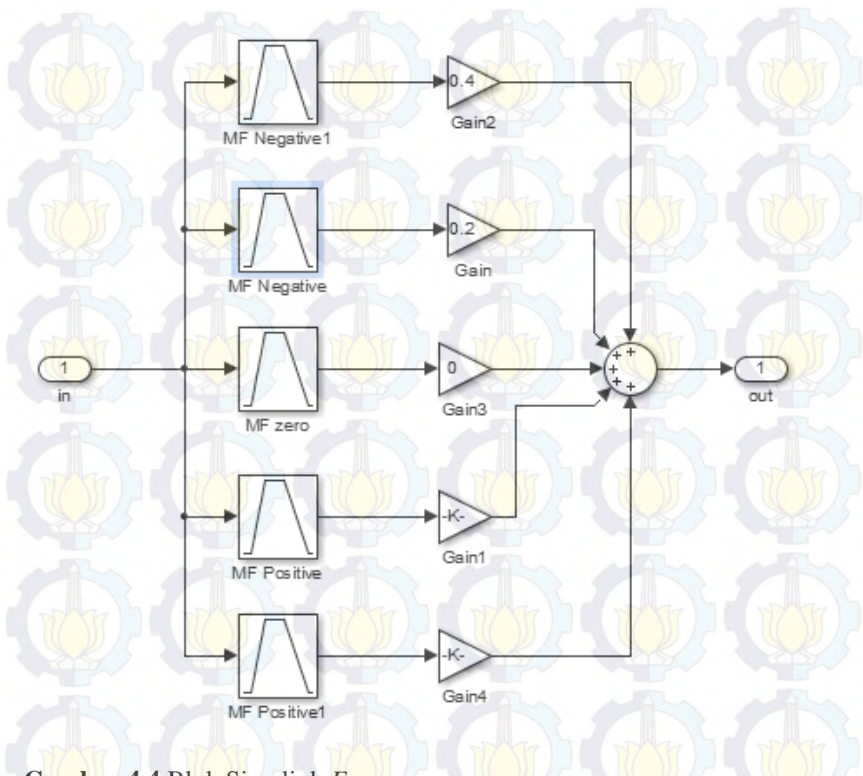
Gambar 4.4 menunjukkan bagian dalam dari blok *fuzzy*. Nilai *error output* menjadi masukan dari lima blok MF dengan fungsi keanggotaan yang telah dijelaskan pada bab sebelumnya. Keluaran blok MF berupa nilai fungsi keanggotaan diberikan *gain* sesuai dengan basis aturan yang telah dibuat. Masing-masing *output gain* kemudian dijumlahkan untuk mewakili operasi *weighted sum*.



Gambar 4.2 Blok Simulink

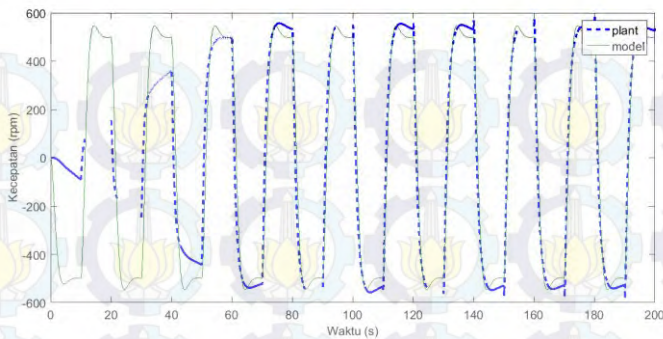


Gambar 4.3 Blok Simulink Filter

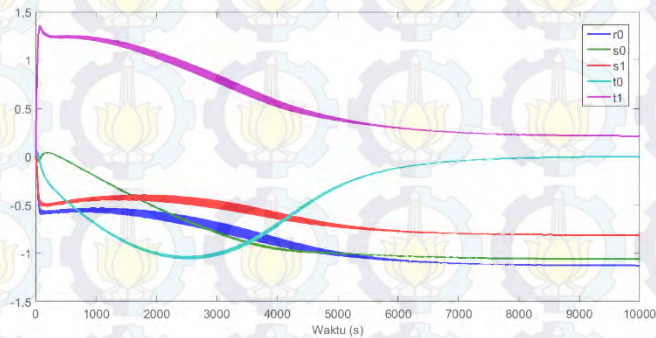


Gambar 4.4 Blok Simulink *Fuzzy*.

Gambar 4.5 menunjukkan bagian dalam dari blok *adjustment*. Blok ini berisi mekanisme penyesuaian parameter kontroler yang telah ditetapkan berupa mekanisme PI sesuai dengan Persamaan 3.2.



Gambar 4.6 *Output Plant* Saat Adaptasi Sampai Detik ke 200



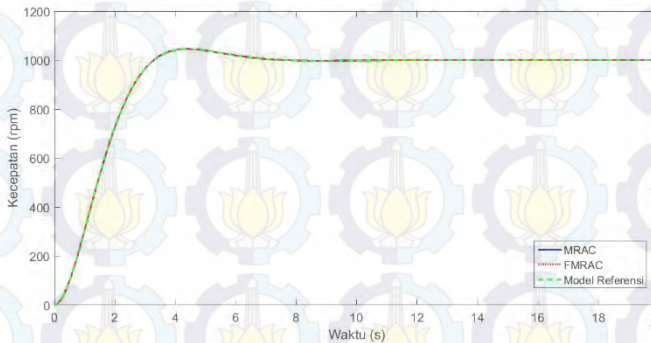
Gambar 4.7 Perubahan Parameter Kontroler

Gambar 4.6 menunjukkan grafik *output plant* saat melakukan adaptasi sampai detik ke 200. Pada kenyataannya, parameter kontroler baru bisa didapatkan dan *output plant* konvergen dengan *output model* setelah 10000 detik waktu simulasi. Ini menunjukkan betapa lamanya proses adaptasi jika hanya menggunakan MRAC biasa.

Gambar 4.5 menunjukkan grafik perubahan parameter kontroler. Parameter kontroler mulai *steady* saat detik ke 6000 dan benar-benar *steady* setelah detik ke 10000. Nilai parameter akhir didapatkan sebagai berikut: $r_0 = 1,058$; $s_0 = 0,9895$; $s_1 = -0,8047$; $t_0 = 0,008181$; $t_1 = 0,2334$.

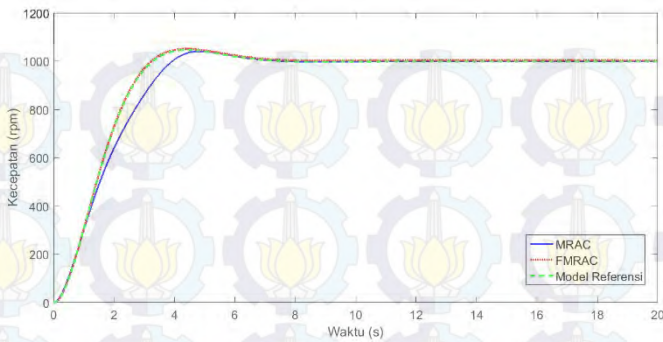
4.4.2.2 *Respon Step*

Gambar 4.8 menunjukkan grafik kecepatan motor saat transien pada kondisi beban nominal. *Input step* yang diberikan adalah 1000 rpm. Seperti yang telah diduga, dari gambar tersebut terlihat bahwa masing-masing kontroler mampu membuat respon *step plant* mengikuti respon *step* model dengan baik.

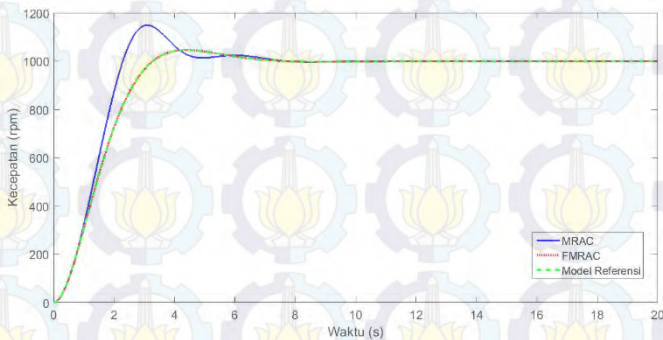


Gambar 4.8 *Step Response* Saat Beban Nominal

Gambar 4.9 dan 4.10 menunjukkan grafik transien kecepatan motor pada kondisi beban maksimal dan minimal. Pada tahap ini, diharapkan sistem melakukan adaptasi nilai parameter kontroler sehingga dapat mengikuti respon model. Dari gambar tersebut terlihat bahwa terdapat *error* antara respon sistem MRAC dengan respon model. Sedangkan respon sistem FMRAC mampu mengikuti respon model dengan baik.

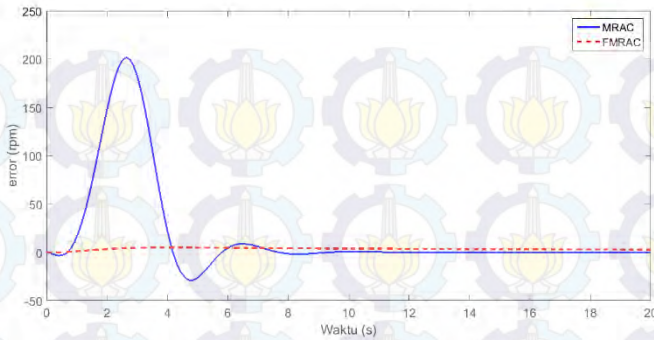


Gambar 4.9 *Step Respon Saat Beban Maksimal*

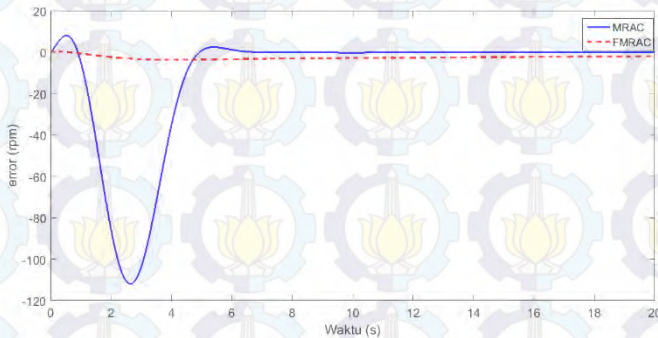


Gambar 4.10 *Step Respon Saat Beban Minimal*

Kemudian Gambar 4.11 dan 4.12 menunjukkan perbandingan *error* antara MRAC dan FMRAC. Terlihat bahwa untuk sistem FMRAC, *error* mendekati nol untuk semua waktu, sementara *error* sistem MRAC lebih besar saat awal simulasi dan mendekati nol setelah sekitar 6 detik.



Gambar 4.11 Error Saat Beban Maksimal



Gambar 4.12 Error Saat Beban Minimal

Tabel 4.2 dan 4.3 menunjukkan kriteria performansi untuk kedua kontroler. *Rise time* (T_R) adalah waktu yang diperlukan agar *output* y dapat naik dari 10% ke 90% dari nilai *steady-state*, sedangkan *settling time* (T_S) adalah waktu yang diperlukan agar *output* y dapat mencapai dan tetap pada rentang $\pm 5\%$ dari nilai *steady state*. Dari tabel terlihat bahwa nilai T_S dan T_R untuk sistem FMRAC hampir sama dengan model dengan selisih masing-masing 0,0147 dan 0,001 untuk beban maksimal dan 0,0228 dan 0,045. Sedangkan untuk sistem MRAC terdapat perbedaan lebih besar yaitu 0,6797 dan 0,594 untuk beban maksimal dan 0,4383 dan 1,18 untuk beban minimal. Kemudian, nilai *root mean square error* (RMSE) dari sistem FMRAC juga jauh lebih kecil dibandingkan dengan

sistem MRAC. Sehingga, dapat dikatakan bahwa sistem FMRAC yang telah dirancang dapat memperbaiki performa transien dari sistem MRAC.

Tabel 4.2 Perbandingan Indeks Performansi Respon *Step* dari MRAC dan FMRAC pada Beban Maksimal

Indeks Performansi	Model referensi	MRAC	FMRAC
T_R (detik)	2,1096	2,7893	2,1243
T_s (detik)	2,949	3,543	2,948
RMSE	0	22,5247	2,6319

Tabel 4.3 Perbandingan Indeks Performansi Respon Transien dari MRAC dan FMRAC pada Beban Minimal

Indeks Performansi	Model referensi	MRAC	FMRAC
T_R (detik)	2,1096	1,5479	2,1324
T_s (detik)	2,949	4,129	2,904
RMSE	0	35,3885	3,6207

4.5 Implementasi Sistem

Pada tahapan ini kontroler yang telah dibuat akan dicoba dijalankan pada *plant real*.

4.5.1 Realisasi *Plant*

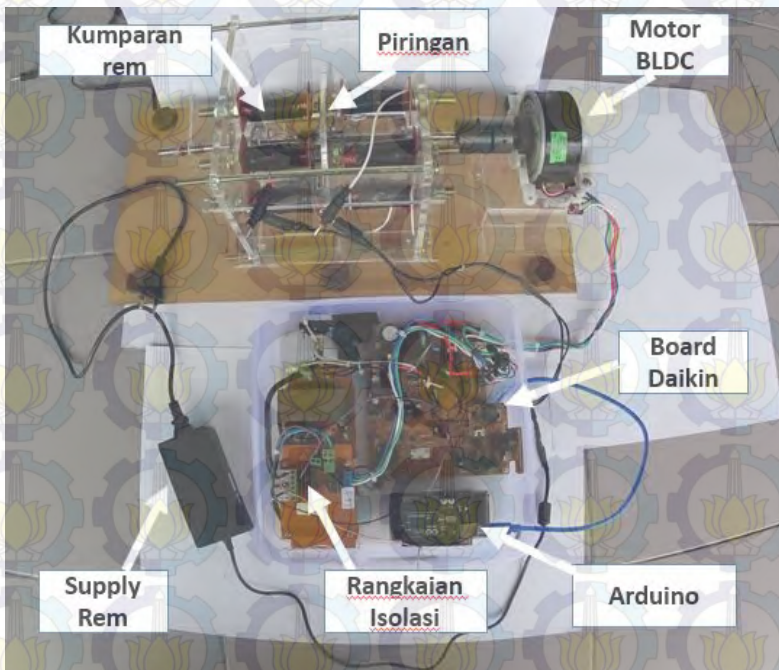
Gambar 4.14 menunjukkan bentuk fisik *Plant* P-1 yang telah kami buat. Seperti yang sudah dibahas sebelumnya, *plant* terdiri dari 2 komponen utama yaitu motor BLDC dan rem elektromagnetik. Poros dari motor diperpanjang dengan sebuah kopling dan disambungkan dengan piringan aluminium. Motor dan kumparan rem, didudukkan pada sebuah *base* yang terbuat dari akrilik.

Board Daikin merupakan *board* yang dibuat oleh perusahaan Daikin untuk kepentingan pengaturan kecepatan motor BLDC. Namun, selain memiliki fungsi pengaturan, *board* ini juga memiliki fungsi sebagai *power supply* yang men-supply daya ke stator dan *driver* motor. Pada *plant* P-1, *board* ini hanya dimanfaatkan fungsi *power supply*-nya, sedangkan pengaturan kecepatannya diambil alih oleh Matlab.

Arduino digunakan sebagai *interface* yang menghubungkan *plant* dengan laptop. Arduino menghitung kecepatan motor dengan cara mengukur periode pulsa yang dikeluarkan oleh *driver* motor dan

kemudian mengirimkan nilai kecepatan tersebut ke laptop. Arduino juga akan menerima masukan data dari laptop untuk mengubah nilai PWM dari pin *output* yang akan digunakan sebagai sinyal kontrol yang mengatur kecepatan motor. Rangkaian isolasi berisi rangkaian *optocoupler* dan *voltage regulator* yang berguna agar pin-pin Arduino tidak terhubung langsung dengan *plant* sehingga memperkecil resiko rusaknya Arduino.

Daya untuk rem elektromagnet di-supply dengan *power supply* tersendiri yang tegangannya dapat diubah-ubah sehingga kekuatan magnetisasi rem juga bisa berubah-ubah.



Gambar 4.13 *Plant* motor BLDC P-1

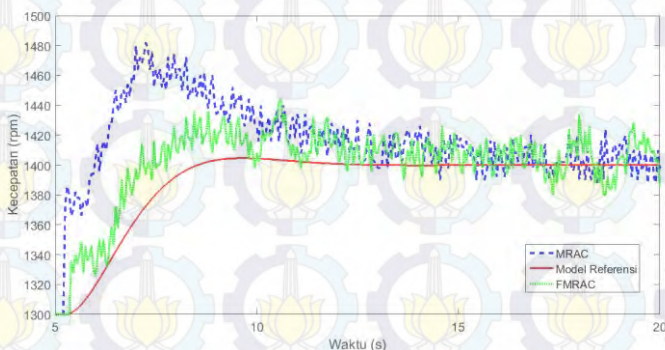
4.5.2 Implementasi Kontroler pada *Plant* P-1

Sama seperti pada tahap simulasi, pada tahap implementasi dilakukan pengujian dengan memberikan *input step* dan kemudian dilihat respon sistemnya. Setelah itu dibandingkan respon sistem yang menggunakan MRAC dengan yang menggunakan FMRAC. Namun,

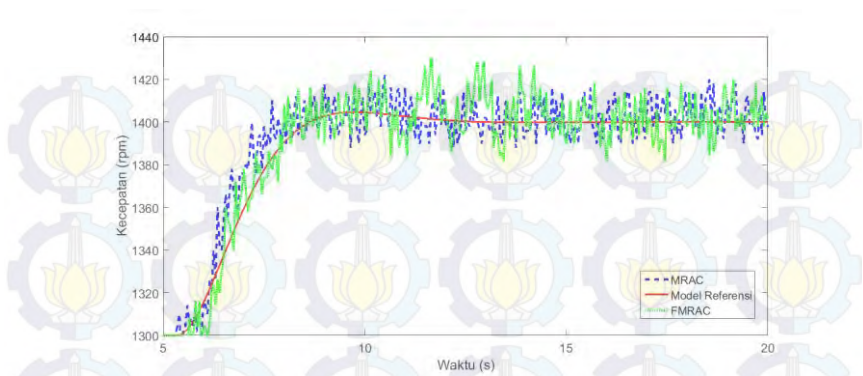
terdapat beberapa pengkondisian yang dilakukan pada tahap implementasi ini. Pertama, nilai *output* pada *rule base* dari *fuzzy inference system* diubah menjadi 0,1 kalinya. Hal ini dikarenakan adanya derau pada *plant real* sehingga jika *output* dari *fuzzy inference system* terlalu besar, maka akan menyebabkan sistem berosilasi.

Pengkondisian kedua berkaitan dengan rentang kerja. Karena rentang kerja yang telah dipilih adalah 1000-1600 rpm, maka nantinya, *input step* yang diberikan akan memiliki nilai awal 1300 dan nilai akhir 1400. Berbeda dengan saat simulasi yang *input step*nya dimulai dari 0. Hal ini dilakukan untuk menghindari pembacaan sensor kecepatan yang kacau saat kecepatan rendah

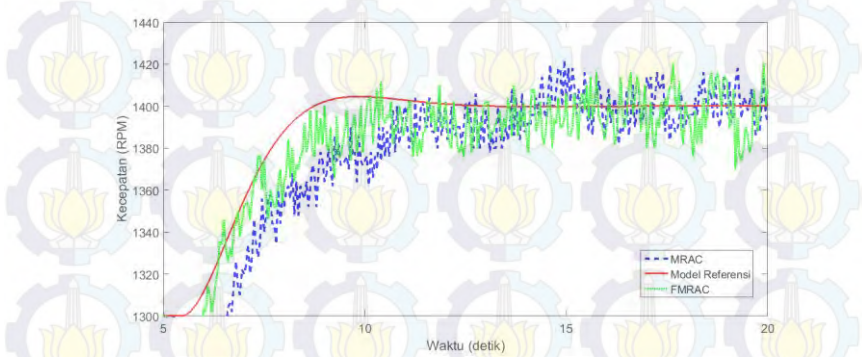
Gambar 4.15, 4.16 dan 4.17 menunjukkan grafik respon transien *output plant* dengan MRAC dan FMRAC pada kondisi beban minimal, nominal dan maksimal. Dari gambar terlihat bahwa, seperti pada simulasi, sistem FMRAC menunjukkan performa yang lebih baik



Gambar 4.14 Step Respon Sistem Saat Implementasi pada Beban Minimal



Gambar 4.15 *Step* Respon Sistem Saat Implementasi pada Beban Nominal



Gambar 4.16 *Step* Respon Sistem Saat Implementasi pada Beban Maksimal

Tabel 4.4, 4.5 dan 4.6 menunjukkan indeks performansi masing-masing sistem MRAC dan FMRAC pada kondisi beban minimal, nominal, dan maksimal. Dari tabel terlihat bahwa selisih *rise time* antara model referensi dan FMRAC selalu lebih kecil yaitu 0,464 untuk beban minimal, 0,005 untuk beban nominal dan 0,259 untuk beban maksimal. Sedangkan selisih *rise time* model referensi dengan MRAC yaitu 1,507 untuk beban minimal, 0,387 untuk beban nominal dan 0,772 untuk beban maksimal. Untuk nilai RMSE, nilai RMSE FMRAC lebih kecil dibanding

MRAC pada beban minimal dan maksimal. Sedangkan pada beban nominal, nilai RMSE MRAC lebih kecil dibandingkan FMRAC.

Tabel 4.4 Indeks Performansi Respon Transien MRAC dan FMRAC Pada Beban Minimal

Indeks Performansi	Model Referensi	MRAC	FMRAC
T_R (detik)	2,15	0,643	1,686
RMSE	0	35,9269	16,0633

Tabel 4.5 Indeks Performansi Respon Transien MRAC dan FMRAC Pada Beban Nominal

Indeks Performansi	Model Referensi	MRAC	FMRAC
T_R (detik)	2,15	1,763	2,145
RMSE	0	8,3255	9,0338

Tabel 4.6 Indeks Performansi Respon Transien MRAC dan FMRAC Pada Beban Maksimal

Indeks Performansi	Model referensi	MRAC	FMRAC
T_R (detik)	2,15	2,922	2,409
RMSE	0	18,6380	15,4448

BAB 5 PENUTUP

5.1 Kesimpulan

Setelah dilakukan pengujian dan analisa dapat ditarik beberapa kesimpulan yaitu:

- a. Dari hasil simulasi dan implementasi, dapat disimpulkan bahwa *Fuzzy-MRAC* dapat memperbaiki waktu konvergensi antara *plant* dengan model dibuktikan dengan selisih *rise-time* dan *settling-time* FMRAC dengan model referensi yang selalu bernilai lebih kecil dibandingkan dengan MRAC biasa.
- b. Pada saat implementasi, didapatkan bahwa nilai RMSE sistem FMRAC saat beban nominal lebih besar dibandingkan dengan MRAC

5.2 Saran

Apabila ada yang ingin menggunakan *plant* P-1 ini untuk dilakukan penelitian lebih lanjut disarankan untuk menambahkan sensor arus untuk rem elektromagnet. Hal ini akan berguna jika sistem pengaturan yang akan diterapkan memerlukan masukan yang merepresentasikan keadaan beban.

LAMPIRAN

Program Arduino

```
int pin = 7;
```

```
int out = 6;
```

```
byte u = 255;
```

```
unsigned long duration;
```

```
unsigned long spd=0;
```

```
word kec;
```

```
byte w1;
```

```
byte w2;
```

```
void setup(){
```

```
  Serial.begin(115200);
```

```
  pinMode(pin,INPUT);
```

```
  pinMode(out,OUTPUT);
```

```
  analogWrite(out,u);}
```

```
void loop(){
```

```
  if (Serial.available()) {
```

```
    u=255-Serial.read();
```

```
    analogWrite(out,u);
```

```
    baca();}}
```

```
void baca(){
```

```
  duration = pulseIn(pin, LOW, 30000);
```

```
  if (duration == 0 && i<3){
```

```
    spd = 0;
```

```
  else{
```

```
    spd=(15*1000000/(2*duration));}
```

```
  kec=int(spd);
```

```
  w1=kec/256;
```

```
  w2=kec%256;
```

```
  Serial.write(w1);
```

```
  Serial.write(w2);}
```

DAFTAR PUSTAKA

- [1] L. K. Baxter, *Capacitive Sensors: Design and Applications*, John Wiley & Sons, 1996.
- [2] S. Banarjee, A. Chakrabarty, S. Mity dan A. Chatterjee, "Feedback Linearizing Indirect Adaptive Fuzzy Control with Foraging based On-line Plant Model Estimation," *Applied Soft Computing*, vol. 11, pp. 3441-3450, 2011.
- [3] H. A. Toliyat dan T. Gopalarathnam, *The Power Electronics Handbook*, CRC Press LLC, 2002.
- [4] H. Hibban, "Desain Kontroler Fuzzy PID Gain Scheduling untuk Pengaturan Kecepatan Motor DC tanpa Sikat," *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, 2015.
- [5] W. J. Brin, "Design and Frabrication of an Eddy Current Brake Dynamometer for Efficiency Determination of Electric Wheelchair Motor," *Thesis*, Fairbor: Wright State University, 2013.
- [6] D. Kho, "Pengertian Optocoupler dan Prinsip Kerjanya," <URL: <http://teknikelektronika.com/pengertian-optocoupler-fungsi-prinsip-kerja-optocoupler/>>, April, 2014.
- [7], "What is a Voltage Follower?," <URL: <http://www.learningaboutelectronics.com/Articles/Voltage-follower/>>, Juli, 2012.
- [8] I. D. L. a. G. Zito, *Digital Control Systems Design, Identification and Implementation*, Paris: Hermes-Lavoisier, 2002.
- [9] J. S. R. Jang, C. T. Sun dan E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Upper Saddle River: Prentice Hall, 1997.

[10] K. J. Astrom dan B. Wittenmark, *Adaptive Control: Second Edition*, Mineola: Dover Publications Inc, 2008.

[11] K. A. Mohideen dan D. K. Valarmathi, "Fuzzy Modified Model Reference Adaptive Controller for Improved Transient Response," dalam *Internasional Conference on Power, Energy and Control (ICPEC)*, 2013.

RIWAYAT HIDUP



Muhammad Iqbal Fauzi lahir di Surabaya tanggal 1 Maret 1993, merupakan anak pertama dari Sunardi dan Faridah. Setelah lulus dari SMA Negeri 2 Pasuruan pada tahun 2011, melanjutkan studi di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember (ITS) Surabaya pada tahun yang sama.